Documented Code For mfirstuc v2.02

Nicola L.C. Talbot

Dickimaw Books

http://www.dickimaw-books.com/

2015-12-17

Contents

1 Introduction

This is the documented code for the mfirstuc package. See `mfirstuc-manual.pdf` for the main user guide.

2 Main Package Code

2.1 Mfirstuc Documented Code

```
1 \NeedsTeXFormat{LaTeX2e}
2 \ProvidesPackage{mfirstuc}[2015/12/17 v2.02 (NLCT)]
```

Requires etoolbox:
```
3 \RequirePackage{etoolbox}
```

\makefirstuc   Syntax:

> \makefirstuc{⟨*text*⟩}

1

Makes the first letter uppercase, but will skip initial control sequences if they are followed by a group and make the first thing in the group uppercase, unless the group is empty. Thus \makefirstuc{abc} will produce: Abc, \makefirstuc{\ae bc} will produce: Æbc, but \makefirstuc{\emph{abc}} will produce *Abc*. This is required by \Gls and \Glspl.

```
4 \newif\if@glscs
5 \newtoks\@glsmfirst
6 \newtoks\@glsmrest
7 \newrobustcmd*{\makefirstuc}[1]{%
```

Check if argument starts with \protect which needs to be skipped.

```
8   \@MFU@p@makefirstuc#1\relax\relax\relax\@end@MFU@protected{#1}%
9 }
```

\@MFU@p@makefirstuc — Internal command to do the required action.

```
10 \def\@MFU@p@makefirstuc#1#2#3\@end@MFU@protected#4{%
11   \ifx\protect#1\relax
12   \@makefirstuc{#2#3}%
13   \else
14   \@makefirstuc{#4}%
15   \fi
16 }
```

\@makefirstuc — Internal command to do the required action.

```
17 \newcommand*{\@makefirstuc}[1]{%
18   \def\gls@argi{#1}%
19   \ifx\gls@argi\@empty
```

If the argument is empty, do nothing.

```
20   \else

21     \def\@gls@tmp{\ #1}%
22     \@onelevel@sanitize\@gls@tmp
23     \expandafter\@gls@checkcs\@gls@tmp\relax\relax
24     \if@glscs
```

Check if we have two sequential control sequences.

```
25       \expandafter\def\expandafter\@gls@tmp\expandafter{\expandafter
26         \ \@gobble #1}%
27       \@onelevel@sanitize\@gls@tmp
28       \expandafter\@gls@checkcs\@gls@tmp\relax\relax
29       \if@glscs\@glscsfalse\else\@glscstrue\fi
30     \fi
31     \if@glscs
32       \@gls@getbody #1{}\@nil
33       \ifx\@gls@rest\@empty
34         \glsmakefirstuc{#1}%
35       \else
36         \expandafter\@gls@split\@gls@rest\@nil
37         \ifx\@gls@first\@empty
```

```
38            \glsmakefirstuc{#1}%
39          \else
40            \expandafter\@glsmfirst\expandafter{\@gls@first}%
41            \expandafter\@glsmrest\expandafter{\@gls@rest}%
42            \edef\@gls@domfirstuc{\noexpand\@gls@body
43              {\noexpand\glsmakefirstuc\the\@glsmfirst}%
44              \the\@glsmrest}%
45            \@gls@domfirstuc
46          \fi
47        \fi
48      \else
49        \glsmakefirstuc{#1}%
50      \fi
51    \fi
52 }
```

Put first argument in `\@gls@first` and second argument in `\@gls@rest`:

```
53 \def\@gls@split#1#2\@nil{%
54    \def\@gls@first{#1}\def\@gls@rest{#2}%
55 }
```

```
56 \def\@gls@checkcs#1 #2#3\relax{%
57    \def\@gls@argi{#1}\def\@gls@argii{#2}%
58    \ifx\@gls@argi\@gls@argii
59      \@glscstrue
60    \else
61      \@glscsfalse
62    \fi
63 }
```

`\@gls@makefirstuc` Make first thing upper case:

```
64 \def\@gls@makefirstuc#1{\mfirstucMakeUppercase #1}
```

`\mfirstucMakeUppercase` Allow user to replace `\MakeUppercase` with another case changing command.

```
65 \newcommand*{\mfirstucMakeUppercase}{\MakeUppercase}
```

`\glsmakefirstuc` Provide a user command to make it easier to customise.

```
66 \newcommand*{\glsmakefirstuc}[1]{\@gls@makefirstuc{#1}}
```

Get the first grouped argument and store in `\@gls@body`.

```
67 \def\@gls@getbody#1#{\def\@gls@body{#1}\@gls@gobbletonil}
```

Scoup up everything to `\@nil` and store in `\@gls@rest`:

```
68 \def\@gls@gobbletonil#1\@nil{\def\@gls@rest{#1}}
```

`\xmakefirstuc` Expand argument once before applying `\makefirstuc` (added v1.01).

```
69 \newcommand*{\xmakefirstuc}[1]{%
70 \expandafter\makefirstuc\expandafter{#1}}
```

\emakefirstuc   Fully expand argument before applying \makefirstuc

```
71 \DeclareRobustCommand*{\emakefirstuc}[1]{%
72   \protected@edef\@MFU@caparg{#1}%
73   \expandafter\makefirstuc\expandafter{\@MFU@caparg}%
74 }
```

\capitalisewords   Capitalise each word in the argument. Words are considered to be separated by plain spaces (i.e. non-breakable spaces won't be considered a word break).

```
75 \newrobustcmd*{\capitalisewords}[1]{%
76   \def\gls@add@space{}%
77   \let\@mfu@domakefirstuc\makefirstuc
78   \let\@mfu@checkword\@gobble
79   \mfu@capitalisewords#1 \@nil\mfu@endcap
80 }
```

```
81 \def\mfu@capitalisewords#1 #2\mfu@endcap{%
82   \def\mfu@cap@first{#1}%
83   \def\mfu@cap@second{#2}%
84   \gls@add@space
85   \@mfu@checkword{#1}%
86   \@mfu@domakefirstuc{#1}%
87   \def\gls@add@space{ }%
88   \ifx\mfu@cap@second\@nnil
89     \let\next@mfu@cap\mfu@noop
90   \else
91     \let\next@mfu@cap\mfu@capitalisewords
92     \let\@mfu@checkword\mfu@checkword
93   \fi
94   \next@mfu@cap#2\mfu@endcap
95 }
96 \def\mfu@noop#1\mfu@endcap{}
```

\mfu@checkword   Check if word should be capitalised. This originally used etoolbox's \ifinlist command but this doesn't work if the word contains grouping.

```
97 \newcommand*\mfu@checkword[1]{%
98   \def\mfu@checkword@arg{#1}%
99   \let\@mfu@domakefirstuc\makefirstuc
100  \forlistloop\mfu@checkword@do\@mfu@nocaplist
101 }
```

\mfu@checkword@do   Handler for loop used by \mfu@checkword

```
102 \newcommand*{\mfu@checkword@do}[1]{%
103 \ifdefstring{\mfu@checkword@arg}{#1}%
104 {%
105   \let\@mfu@domakefirstuc\@firstofone
106   \listbreak
107 }%
108 {}%
109 }
```

\@mfu@nocaplist	List of words that shouldn't be capitalised.

```
110 \newcommand*{\@mfu@nocaplist}{}
```

\MFUnocap	Provide the user with a means to add a word to the list.

```
111 \newcommand*{\MFUnocap}[1]{\listadd{\@mfu@nocaplist}{#1}}
```

\gMFUnocap	Global version.

```
112 \newcommand*{\gMFUnocap}[1]{\listgadd{\@mfu@nocaplist}{#1}}
```

\MFUclear	Clear the list

```
113 \newcommand*{\MFUclear}{\renewcommand*{\@mfu@nocaplist}{}}
```

\xcapitalisewords	Short-cut command:

```
114 \newcommand*{\xcapitalisewords}[1]{%
115   \expandafter\capitalisewords\expandafter{#1}%
116 }
```

\ecapitalisewords	Fully expand argument before applying \capitalisewords

```
117 \DeclareRobustCommand*{\ecapitalisewords}[1]{%
118   \protected@edef\@MFU@caparg{#1}%
119   \expandafter\capitalisewords\expandafter{\@MFU@caparg}%
120 }
```

2.2 Mfirstuc-english Documented Code

```
121 \NeedsTeXFormat{LaTeX2e}
122 \ProvidesPackage{mfirstuc-english}[2015/09/09 v2.0 (NLCT)]
```

Load mfirstuc if not already loaded:
```
123 \RequirePackage{mfirstuc}
```

Add no-cap words. This list isn't complete. I don't intend adding disputed words. Additional words can be added to the document in a similar manner.
```
124 \MFUnocap{a}
125 \MFUnocap{an}
126 \MFUnocap{and}
127 \MFUnocap{but}
128 \MFUnocap{for}
129 \MFUnocap{in}
130 \MFUnocap{of}
131 \MFUnocap{or}
132 \MFUnocap{no}
133 \MFUnocap{nor}
134 \MFUnocap{so}
135 \MFUnocap{some}
136 \MFUnocap{the}
137 \MFUnocap{with}
138 \MFUnocap{yet}
```

Change History

Index

Numbers written in italic refer to the page where the corresponding entry is described; numbers underlined refer to the code line of the definition; numbers in roman refer to the code lines where the entry is used.